What is Claimed is:

[c1] A method of comparing a query dataset N with a subject dataset M, comprising:

dividing said query dataset N into n_N data elements having a size within a specified range;

dividing said subject dataset M into n_M data elements having a size within said specified range;

determining a number of tasks for an entire comparison of datasets N and M as $n_{N \times} n_M$;

sending all data elements and task definitions to a master CPU of a master-slave distributed computing platform,

wherein task definitions comprise at least one comparison parameter, at least one executable element capable of performing comparisons, a query data element ID/descriptor, and a subject data element ID/descriptor, and

wherein data elements are sent alternately from query and subject data elements;

sending a task definition for each task from the master CPU to one of a plurality of slave CPUs when all parts of a task definition and data elements referenced by said task definition are available at said master CPU;

sending data elements referenced by said task definition to said slave CPU;

performing each task on a slave CPU; and

returning task results for each task to said master CPU.

[c2] The method of claim [c1], further comprising randomizing sequence order of each dataset if either dataset contains related sequences in a contiguous arrangement.

[c3] The method of claim [c1], further comprising formatting said datasets so as to use exactly the same ambiguity substitutions.

[c4] The method of claim [c1] wherein dividing said datasets into data elements further comprises:

stripping all metadata from data;

packing said data into an efficient structure;

creating an index for said data and packing said index and said data in an uncompressed data structure; and

compressing said uncompressed data structure into a data element using a redundancy reduction data compression method.

[c5] The method of claim [c1], further comprising sending remaining data elements from a more numerous of said datasets to said master CPU followed by all task definitions for otherwise complete tasks if there are fewer data elements from one dataset.

[c6] The method of claim [c1] wherein performing a task on said slave CPU further comprises:

uncompressing and unpacking data from said query and subject data elements;

looping through query sequences from said query data element to perform setup, preprocessing and table generation for each row of comparisons;

looping through subject sequences from said subject data element and, for each pair of query and subject sequences, performing a comparison using said executable element and finding results based on said at least one comparison parameter; and

storing minimal information that will allow reconstruction of said result.

[c7] The method of claim [c6] wherein storing said minimal information comprises:

storing index information for said query and said subject sequence;

storing bounds information for start and stop of said query and subject sub sequences;

storing data that quantify fulfillment of significance criteria for a significant match; and

storing an efficiently encoded representation of alignment between said bounds corresponding to a high-scoring segment pair.

[c8] The method of claim [c7], further comprising storing a seed point and sum-set membership for each alignment for BLAST.

[c9] The method of claim [c7], further comprising storing task results in a task result file, said file including query and subject sequence data and metadata corresponding to the task that the results came from, metadata for the subject sequence, the partial subject sequence data corresponding to the subject bounds of the significant alignment result, and any other results data for each result in the task results.

[c10] The method of claim [c9], further comprising generating a BLAST report for each query data element.

[c11] The method of claim [c10], further comprising concatenating results from all BLAST reports to produce a text file identical to a blastall run of said query and subject datasets.

[c12] The method of claim [c1] wherein said datasets are selected from the group consisting of genomic and proteomic databases.

[c13] A system for comparing a query dataset N with a subject dataset M, comprising:

a master CPU of a master-slave distributed computing platform;

a plurality of slave CPUs capable of communication with said master CPU; and

a client CPU with instructions for:

dividing said query dataset N into n_N data elements having a size within a specified range;

dividing said subject dataset M into n_M data elements having a size within said specified range;

determining a number of tasks for an entire comparison of datasets N and M as $n_N \times n_M$;

sending all data elements and task definitions to said master CPU of a master-slave distributed computing platform,

wherein task definitions comprise at least one comparison parameter, at least one executable element capable of performing comparisons, a query data element ID/descriptor, and a subject data element ID/descriptor, and

wherein data elements are sent alternately from query and subject data elements;

said master CPU comprising instructions for:

sending a task definition for each task to one of said plurality of slave CPUs when all parts of a task definition and data elements referenced by said task definition are available at said master CPU; and

sending data elements referenced by said task definition to said slave CPU; and

said slave CPUs including instructions for:

performing each task; and

returning task results for each task to said master CPU.

[c14] The system of claim [c13], further comprising means for randomizing sequence order of each dataset if either dataset contains related sequences in a contiguous arrangement.

[c15] The system of claim [c13], further comprising means for formatting said datasets so as to use exactly the same ambiguity substitutions.

[c16] The system of claim [c13], wherein said instructions for dividing said datasets into data elements further comprises instructions for:

stripping all metadata from data;

packing said data into an efficient structure;

creating an index for said data and packing said index and said data in an uncompressed data structure; and

compressing said uncompressed data structure into a data element using a redundancy reduction data compression method.

[c17] The system of claim [c13], further comprising instructions for sending remaining data elements from a more numerous of said datasets to said master CPU followed by all task definitions for otherwise complete tasks if there are fewer data elements from one dataset.

[c18] The system of claim [c13], wherein instructions for performing a task on said slave CPU further comprises instructions for:

uncompressing and unpacking data from said query and subject data elements;

looping through query sequences from said query data element to perform setup, preprocessing and table generation for each row of comparisons;

looping through subject sequences from said subject data element and, for each pair of query and subject sequences, performing a comparison using said executable element and finding results based on said at least one comparison

parameter; and

storing minimal information that will allow reconstruction of said result.

[c19] The system of claim [c18], wherein said instructions for storing said minimal information comprises instructions for:

storing index information for said query and said subject sequence;

storing bounds information for start and stop of said query and subject sub sequences;

storing data that quantify fulfillment of significance criteria for a significant match; and

storing an efficiently encoded representation of alignment between said bounds corresponding to a high-scoring segment pair.

[c20] The system of claim [c19], further comprising instructions for storing task results in a task result file, said file including query and subject sequence data and metadata corresponding to the task that the results came from, metadata for the subject sequence, the partial subject sequence data corresponding to the subject bounds of the significant alignment result, and any other results data for each result in the task results.

[c21] The system of claim [c20], further comprising instructions for generating a BLAST report for each query data element.

[c22] The system of claim [c21], further comprising means for concatenating results from all BLAST reports to produce a text file identical to a blastall run of said query and subject datasets.

[c23] The system of claim [c13], wherein said datasets are selected from the group consisting of genomic and proteomic databases.